CONNEXIONS™

The Interoperability Report

January 1989 Volume 3, No. 1

ConneXions—
The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published by Advanced Computing Environments, 480 San Antonio Road, Suite 100, Mountain View, CA 94040, USA. Phone: 415-941-3399.

ISSN 0894-5926

From the Editor

When the Internet Protocol Suite was conceived, the designers settled on a 32 bit IP addressing scheme. The enormous growth in the number of organizational networks, particularly based on LAN technology has rendered the 32 bit addresses somewhat awkward to use in a "flat" sense. (If the designers were to meet again today, we would probably end up with a larger address field such as the 40 octets used in OSI). Since changing the IP addressing scheme is not something one does overnight, the research community focused its attention on how to use the existing system in a "clever" way. The resulting method known as *subnetting* is the subject of this special issue. In three articles we will explore subnetting from both a theoretical and practical perspective. The material in these articles was first presented at INTEROP™ 88.

The first article is by Jeff Mogul of DECWRL and explains the concept of subnets. Jeff co-authored the RFC which specifies the Internet Standard Subnetting Procedure. The second article is by Barry Shein of Encore Computer and gives some hints for implementors and network administrators who plan to set up subnetting. The third article is by Fred Ball of Ford Motor Company. Fred gives an overview of how various subnetting schemes can be used to configure large complex networks.

RFC 1083 entitled ***IAB Official Protocol Standards*** was issued in December. This memo describes the state of standardization of protocols used in the Internet as determined by the Internet Activities Board (IAB). An overview of the standards procedures is presented first, followed by discussions of the standardization process and the RFC document series. An explanation of the terms is presented, and the lists of protocols in each stage of standardization follows. Finally, pointers to references and contacts for further information is given. This memo will be issued quarterly. Please be sure the copy you are reading is dated within the last three months. *Do not use this memo after 31-March-1989.*

UC Berkeley recently announced the availability of a new networking software package. For details, see page 19.

The US domain is part of the Domain Name System. Jon Postel gives an overview of the purpose and operation of this domain on page 20.

Letters to the Editor are always welcome. In response to an article in our October 1988 issue, Helmut Hildebrandt of the Stuttgart Institute of Technology reflects on upper layer functionality.

Finally, we review a couple of the relevant books in the networking field. More of these reviews will follow throughout 1989.

Subnetting: A Brief Guide

by Jeffrey Mogul, Digital Equipment Corp.,
Western Research Laboratory

Introduction

Hosts addresses in the Internet Protocol (IP) are 32 bits wide. In the original specification of IP, RFC 791 [9], these addresses were divided into two fields, called the *network* field and the *host* field. This creates a two-level hierarchy, with two benefits: (1) routing tables in the gateways (and in the routing protocols) need only contain routes to each network, not to each host, and (2) host addresses may be assigned by local administrations, rather than by a central site.

When IP was designed, the full implication of LAN technology was not yet obvious. It was not even apparent that there would be more than a few hundred networks, nor did people imagine just how many hosts an organization would try to put on a network.

What actually happened is that large organizations, such as universities and corporations, began to create large organizational networks. Because of geographical and administrative constraints, and because bridges (link-level connections between LANs) are not acceptable in many cases, these networks consist of many LANs connected with gateways (network-level routers). There was an explosive growth in the number of networks.

Assigning each of these LAN segments its own IP network number leads to two serious problems: (1) the campus network management would have to ask the Internet Numbers Czar for a new network number whenever a new segment is installed, and (2) the size of internetwork routing tables would grow tremendously. Since gateway storage space is at a premium, and since the existing routing exchange protocols do not work well for large numbers of networks, the number of IP networks could not be allowed to grow so explosively. Moreover, the value of listing all the segments of a campus network in the world-wide routing table is almost nil; the route between hosts on two campuses is almost certain to be the same no matter what LAN segments those hosts reside on.

A solution to the network explosion

The solution was to add another level of hierarchy to the IP addressing structure. Instead of the two-level (*network, host*) hierarchy, we now have a three-level (*network, subnet, host*) hierarchy. Each organization is assigned one, or at most a very few, network numbers. An organization is then free to assign a distinct subnet number to each of its internal LANs and WANs; host numbers are then assigned on each subnet.

The subnet structure of a network is *never* visible outside that network; from a host (or gateway) anywhere else in the Internet, the route to that network is the same no matter what subnet the destination host is on. Some people have complained that this reduces the efficiency and flexibility of the routing mechanism; this is true in theory, but seldom in practice. On the other hand, the benefits are clear: an organization is free to administer its internal network segments without having to deal with the Internet management, and the size of internetwork routing tables is still manageable.

Conceptually, the IP address is treated as consisting of a *network* field and a *local-part* field. Each organization is allowed to choose how to divide up the local-part field of its IP addresses into subnet and host fields. (An organization that does not divide its network into subnets simply allocates no bits to the subnet field.) Normally, the least significant bits are used for the host number, and the bits between the host number and the network number are used for the subnet number. For example, Stanford University (Class A network number 36) has 24 bits to divide up, using 8 bits for the subnet number (Stanford has about 74 subnets) and 16 bits for the host number. Using the usual "dotted quad" notation, host 3 on subnet 17 on network 36 would have the address 36.17.0.3.

For reasons mostly of administrative convenience, it is typical to divide up the address on eight-bit boundaries. For Class A networks, this means either 8 bits of subnet and 16 bits of host, or 16 bits of subnet and 8 bits of host (useful if an organization is widely dispersed). For Class B networks, this means 8 bits for both subnet and host numbers. For Class C networks, with only 8 bits to divide, or for organizations that need to balance more delicately between the number of subnets and the maximum number of hosts per subnet, the subnet field may be any number of bits wide.

In some organizations, it is necessary to accommodate both a large number of networks, and a few networks with a large number of hosts. This requires the use of more than one address structure within the organization; we discuss this in more detail later on. It is important to realize, however, that hosts on other networks need know nothing about the division in use, no matter what it is.

Internet Standard Subnetting Procedure

The standard for implementing IP subnetting is RFC 950 [7]. Anyone interested in actually doing an implementation should of course consult the RFC; this section covers the important concepts at the level necessary for understanding how to manage a network.

The first concept is that hosts and gateways have very different requirements. (Many people attempt to use as a gateway software that is only suited for use as an IP host; this can be a serious mistake. See RFC 1009 [1] for a discussion of the requirements for IP gateways.) A gateway must know how to route any packet that it receives; the details of how to do this are quite complex and are beyond the scope of this article. A host must know only two things: (1) is the destination of a datagram on my directly-connected network, and (2) if it is not, what is the address of some gateway on my directly-connected network.

Since every gateway must be able to route any packet, and since all hosts are required to support ICMP Redirect messages [10], once a host has determined that a datagram must go via a gateway, it does not matter which gateway the datagram is sent to. That gateway will both forward the packet properly, and, if the gateway is not the appropriate first hop, it will send an ICMP Redirect Message back to the source host. The source host must update its routing cache to reflect the ICMP Redirect, after which all datagrams to the destination will go immediately to the right gateway. In other words, hosts are not expected to understand the topology of the Internet, and *hosts are not supposed to make routing decisions.*

continued on next page

Subnetting: A Brief Guide *(continued)*

We must leave open the question of how a host knows the addresses of its neighboring gateways; unfortunately, there is no standard mechanism for gateway discovery, so this varies from implementation to implementation, and has nothing to do with subnetting.

The remaining question is how a host determines if a destination address is "local" or "remote." Before subnetting, this was done by extracting the network number fields from the destination address and the host's own address; if they were equal, then the destination was local. With subnetting, it is not this simple, since the destination might be on a different subnet of the same network.

Address mask

This brings us to the second important concept, the "Address Mask." The Address Mask is a 32-bit wide bit-mask that separates the host number field from the subnet number and network number fields. In the simplest case, there is one Address Mask for every network (later on, we will describe an exception).

The Address Mask has "1" bits in the positions corresponding to the network number and subnet number fields, and "0" bits elsewhere (the host number field). A host uses the mask by doing a bitwise-AND between the mask and the destination address, and another bitwise-AND between the mask and its own address. If the resulting values are equal, then the destination address is clearly on the same subnet of the same network as the host, and it is therefore a local address. If the values do not match, then the destination may either be on a different network altogether or on a different subnet of the same network; from the point of view of the host, *it does not matter which*, because in either case the datagram will have to be sent via a gateway.

For example, the Address Mask for the Stanford University network described above is 255.255.0.0; the first octet (byte) of "1" bits corresponds to the Class A network number, and the second octet of "1" bits corresponds to the 8-bit wide subnet number field.

Discovering the address mask

Clearly, a host must have some way to discover the appropriate Address Mask value. In many systems, this can be read from a configuration file at boot time (in a 4.xBSD-derived UNIX system, for example, the value could be set in `/etc/rc.local`). Because not all hosts use configuration files, there is a mechanism by which a host can discover the Address Mask over the network instead.

This is the purpose of the ICMP Address Mask Request message, described in detail in RFC 950. A host can broadcast the request on its directly-connected network, or it can send the request directly to one of its neighbor gateways, if it knows the address. (Either of these can be done without first knowing the Address Mask.) A gateway receiving this request sends an ICMP Address Mask Reply message, containing the Address Mask in effect on this subnet. If there is no gateway to respond to the request, then there is no chance of reaching a non-local host anyway, so this is not a disaster.

Subnetting and broadcasts

Although IP did not initially include the concept of a broadcast address, this turned out to be an important feature and a standard was subsequently specified in RFC 919 [5].

When subnetting is not in use, there are two possible broadcast addresses: the "wildcard" or "all ones" address, 255.255.255.255, and the one consisting of the network number, with the remaining bits all "1"s (e.g., on net 36 the broadcast address would be 36.255.255.255).

Because an organization might conceivably want to distinguish between broadcasts to a single subnet and broadcasts to the entire organization, a third possible broadcast address is allowed when subnets are in use. This *subnet broadcast address* consists of the network number and subnet number fields as for a normal host address, and the host number field set to all "1"s, e.g., on subnet 8 of network 36, the subnet broadcast address would be "36.8.255.255."

Although a standard was proposed for the implementation of organization-wide broadcasting over a subnetted network [6], it has never been adopted and probably has never been implemented. As a result, all three broadcast addresses have the same effect: the broadcast is sent only on the directly-connected subnet. A program running on a multi-homed host (a host connected to more than one subnet) will probably want to use the subnet-specific broadcast addresses, to make sure that the broadcasts go to the right subnets, but for most applications the 255.255.255.255 address is sufficient. Note, however, that a host must be prepared to accept datagrams sent to any of the three possible broadcast addresses.

Inhomogeneous subnetting

As mentioned earlier, some organizations have both a large number of subnets and a few subnets with a large number of hosts. Especially since Class A network numbers are scarce, such organizations may not be able to divide up the 16 bits remaining in a Class B IP address in a convenient manner. Any division that allocates enough bits to represent a large enough number of subnets may not leave sufficient bits to represent a large enough number of hosts. For example, if organization XYZ, with a Class B network, has 500 subnets with 10 hosts and one subnet with 1000 hosts, there is no fixed division of the address that will work.

It has been proposed that, in such cases, different Address Masks be used on different subnets of the same organization. For example, organization XYZ might use a mask of 255.255.255.0 on 126 of its subnets, allowing 254 hosts on each of those networks, and a mask of 255.255.128.0 on one other subnet, allowing 32767 hosts on that subnet. (At least one bit that is present in the subnet number fields of all subnets must be used to distinguish between subnets using the two different masks, since otherwise there would be an ambiguity in interpreting the results.)

One application of inhomogeneous subnetting is to create additional levels of addressing hierarchy. Conceptually, a network is divided into subnets, some of which are in turn divided into sub-subnets. This would allow the fine structure of routing information for one subnet to be hidden from gateways in a different subnet. This is probably not useful; few organizations are complex enough to warrant the creation of yet another level of routing protocol.

As of this writing, there is not yet a standard in place for inhomogeneous subnetting. Such a standard would specify how the reserved bit (or bits) of the subnet number encodes which Address Mask is appropriate, and how routing protocols are affected.

continued on next page

Subnetting: A Brief Guide *(continued)*

Also, the use of inhomogeneous subnetting creates complications for implementations of gateways and of multi-homed hosts.

Organizations that need to use this mechanism should proceed carefully. Most organizations would do better to split their large networks with gateways; this not only avoids the issue of inhomogeneous subnetting, but it can also simplify network administration by limiting the number of hosts that can be affected by a problem with one subnet.

When to not use subnetting

Clearly, an organization with one LAN, containing a small number of hosts and covering a small area, does not need to use subnetting. Similarly, at some point an organization may grow to be so large that it runs out of address space; at this point, it must either obtain a higher class of network number, or split into two networks (each internally subnetted, of course).

Note that an organization that starts out small should still be prepared to start using subnets, since networks inevitably grow. If the appropriate host software is chosen, switching to subnetting should simply require changing a few lines in configuration files.

Some organizations are so geographically dispersed that the cost of routing all external traffic via the same path becomes untenable. For example, consider a company with two offices, on the either coast of the U.S., connected via a low-speed network. Assume that the gateway between this company's network and the "outside world" is located in San Francisco; now suppose that a host at a university in Boston wants to send a datagram to a host of this company that is also in Boston. The packets would have to traverse the continent twice, adding considerable delay and cost. In effect, the assumption that the internal (subnet) structure of this company is unimportant to outsiders is incorrect.

There are two solutions to this problem. One is for the company to install a high-speed link between its office; this does not eliminate the need for two cross-country hops, but it does make them a lot less painful. The other solution is to use different network numbers for the two offices. This would allow the university host in Boston to route directly to the company office in Boston. Transcontinental traffic internal to the company could still be routed along the link controlled (and paid for) by the company, so it would be no more expensive. Thus, in this case it may not be a good idea to treat all the LANs of the company as subnets of a single network.

Choice of subnetting parameters

There are a few rules that must be followed in assigning subnet numbers and host numbers. Simply put, neither the subnet number nor the host number may be either all "0" bits or all "1" bits. In the common case of an 8-bit subnet field and an 8-bit host field, this means that subnet numbers 0 and 255 are illegal, as well as host numbers 0 and 255. The reason for this is quite simple: these are values that are used, or have been used, to denote either broadcasting [5] or "unknown." Assigning these values to actual hosts or subnets is bound to lead to trouble.

A simple rule of thumb has been suggested for assigning host and subnet numbers [11]:

- Use the least significant bits of the IP address for the host number field.
- Allocate host numbers starting at 1 and increasing in sequence.
- To allocate a subnet number, start an auxiliary counter at 1 and increase it in sequence. Subnet numbers are formed by reversing the bits of this counter, so that the high-order bits are set first. For example, if the subnet number field is 8 bits wide, the first few subnet numbers assigned would be 128, 64, 192, 32, 160, 96, etc.

The effect of this rule is to leave zero bits near the boundary between the host and subnet number fields for as long as possible. Then if you must move this boundary, perhaps because you install more subnets than you originally expected, you may not need to change the addresses you have already assigned.

When practical, it is best to divide up the IP address into fields that are aligned on octet boundaries. This makes it much easier to decipher an address expressed in the usual dotted quad notation, a task which is surprisingly common in the administration of a network. Also, octet-aligned subnet fields simplify the use of the IN-ADDR.ARPA domain, used to support translations from IP address back to host names [3]; this is especially true if you want to operate separate naming domains on specific subnets [4].

One interesting special case is the use of subnetting to identify point-to-point links. In this situation, each point-to-point link could have its own subnet number; the two hosts at either end of the link have individual host numbers. If this scheme is used, the host number field need be only 1 bit wide; the rest of the local-part field can be used for the subnet number. This is most useful for networks comprised solely of point-to-point links.

This is not the only way to manage point-to-point links; one can also connect several point-to-point links to a single gateway, give them all the same subnet number, and let the normal routing mechanisms lead datagrams for that "subnet" to the gateway. The gateway, in turn, must use host-specific routes to each its neighbors on the point-to-point links. The subnet number field should probably be octet-aligned, unless there will be many of these star-configuration gateways.

Misconceptions

One area of confusion is over the connection between subnetting and domain naming [3]. Both both have something to do with identifying hosts, both involve imposing a hierarchy, and both involve decentralized administration, but there is actually very little connection between the structure of IP addresses and domain names. A single domain may contain names for hosts on many different subnets (or even on different networks); a single subnet may contain hosts that are named in several different domains.

The one connection is the IN-ADDR.ARPA domain, as mentioned earlier. Since the names in this special domain break at what correspond to the octet boundaries of an IP address, where octet-aligned subnet fields are used, the corresponding subnets show up as distinct subdomains of IN-ADDR.ARPA [4].

continued on next page

Subnetting: A Brief Guide *(continued)*

Another common misconception is that it is possible to separate two subnets of one network, connecting them only over a distinct IP network. This is tempting if you are a geographically dispersed organization, but it will not work: since the subnet structure of one network must be invisible everywhere outside that network, there is no way for a gateway in the middle to know how to route packets for the disconnected subnets.

Implementation issues

As mentioned earlier, anyone actually planning to implement subnetting should read RFC 950. It may help to know something about implementation issues, however, in order to understand subnetting in general.

Subnet implementation can be divided into two categories: host issues and gateway issues. Host IP software must have a means of obtaining the Address Mask, and must implement ICMP Redirect messages; that is, a host must do the right thing when it receives a Redirect. Beyond that, the subnet-related per-datagram processing for outgoing datagrams is quite simple, consisting simply of the "local"/"remote" decision. For incoming datagrams, the only issue is to recognize the subnet-specific broadcast address in addition to the network-specific and wildcard broadcast addresses.

Gateway IP software must be able to extract the subnet field from the destination IP address of a packet it is forwarding, and use this subnet field to choose a route for the packet. Therefore, a gateway must also implement a routing protocol that understands about subnets. Routing protocols are quite complex and are beyond the scope of this article.

Because subnetting was added to the IP specification after a large number of hosts were already on the Internet, there were (and still are, alas) many hosts on subnetted networks that did not implement subnetting. Such a host could reach destinations on its own subnet, and destinations on remote networks, but not destinations on remote subnets of the same network. This is because the host would believe the destination address to be "local" and attempt to send to it directly, when it must actually be reached via a gateway.

Proxy ARP

A mechanism known variously as "Proxy ARP," "Promiscuous ARP," "The ARP Kludge," or "The ARP Hack" was invented to avoid this problem on LANs such as Ethernets, where ARP (the *Address Resolution Protocol* [8]) is already in use. Since a non-subnetting host on a subnetted Ethernet attempts to use ARP to find the Ethernet address of a host that it believes is "local," in Proxy ARP a gateway hearing an ARP request for a host that it knows is "remote" simply responds to that ARP request with its own Ethernet address. The dumb host then sends the datagrams to that address, which is what it would do anyway if it knew that the destination was remote.

Proxy ARP requires some changes to gateway software, is not a solution for non-ARP LANs, and in any case should not be treated as a permanent replacement for implementation of subnetting. For example, if the route changes the host may receive ICMP Redirect messages but may not know what to do with them, or the host may choose an inappropriate datagram size, resulting in potentially harmful fragmentation [2].

A small bug in the RFC

In section 2.2 of RFC 950 (entitled "Changes to Host Software to Support Subnets"), there is some pseudo-code that indicates how host software must be changed to support subnetting. There is a small bug in this pseudo-code; it implies that if destination of a datagram is remote, then the destination address should be ANDed with the Address Mask before being used to select a gateway.

This is actually ridiculous, since if the destination is remote, it may not even be on the same network as the source host, and consequently combining the source host's Address Mask with this destination address will give meaningless results. Instead, the entire destination address must be used when selecting a gateway; this is especially important because host-Redirect entries may be present in the routing cache.

Acknowledgements

I would like to thank Fred Ball, Bob Braden, Sergio Heker, J. Q. Johnson, and Barry Shein for helpful comments they made during the preparation of this article.

References

[1] Braden, R. & Postel, J., "Requirements for Internet Gateways," RFC 1009.

[2] Kent, C.A. & Mogul, J., "Fragmentation Considered Harmful," Report 87/3, DEC Western Research Laboratory, November, 1987. Expanded version of paper presented at SIGCOMM '87.

[3] Mockapetris, P.V., "Domain Names—Implementation and Specification," RFC 1035.

[4] Mockapetris, P.V., "Domain name/subnet non-relationship," Electronic distribution of the TCP-IP Discussion Group, Message-ID <8806292148.AA25642@venera.isi.edu>.

[5] Mogul, J., "Broadcasting Internet Datagrams," RFC 919.

[6] Mogul, J., "Broadcasting Internet Datagrams in the Presence of Subnets," RFC 922.

[7] Mogul, J. & Postel, J.,"Internet Standard Subnetting Procedure," RFC 950.

[8] Plummer, D. C., "An Ethernet Address Resolution Protocol," RFC 826.

[9] Postel, J., "Internet Protocol," RFC 791.

[10] Postel, J., "Internet Control Message Protocol," RFC 792.

[11] Zeilenga, K., "Allocating subnets," Electronic distribution of the TCP-IP Discussion Group, Message-ID <23603@hi.unm.edu>.

JEFFREY C. MOGUL received an S.B. from M.I.T. in 1979, an M.S. from Stanford in 1980, and his PhD from the Stanford Computer Science Department in 1986. While at Stanford he produced the distribution of the Stanford implementation of PUP protocol software for UNIX. He is author or co-author of a number of RFCs, including those specifying Internet Standards for subnets, broadcasting, and RARP. Since 1986, he has been a researcher at the Digital Equipment Corporation Western Research Laboratory, working on network and operating systems issues for high-performance computer systems.

Subnet Implementation

by Barry Shein, Encore Computer Corp.

How a host processes an outgoing datagram

The IP header contains the destination IP address. There are *two* possible decisions to be made, send the datagram to a host on the local network or send it to gateway destined for another network. There may be more than one gateway reachable from the host, so which gateway to send to may have to be determined.

The IP address in the datagram is ANDed with the subnet mask and the result is compared with the host's own address ANDed with the same subnet mask. If the result of those two operations are equal then the host is on the local network and the datagram may be sent directly. If not then the routing layers must be consulted for an appropriate gateway. This is usually resolved either by discovering a default gateway or searching through a list of routes in an internal routing table (possibly concluding that the datagram should go to the default gateway anyhow.)

A host might have more than one hardware interface, each interface will have its own IP address and possibly its own subnet mask (some earlier UNIX systems were unable to store a per-interface subnet mask, a real problem.) Of course, not every interface has to be subnetted. The solution is the same as above except that each interface is iterated through to see if the datagram's destination is local.

Configuration

In the simple case of adding a host with one interface to an already subnetted network only two things are needed, to choose an address on the local subnet for the host and to set up the system startup files to reflect that address and the subnet mask being used. On a UNIX system this is done in an `/etc/rc` (Typically `/etc/rc.local` or `/etc/rc.boot`) file with a command like:

```
/etc/ifconfig ie0 bu-cs.bu.edu netmask 255.255.255.0 -trailers up
```

The host is BU-CS.BU.EDU (whose corresponding numeric address will be looked up in the host table) and the netmask is 255.255.255.0.

BU-CS.BU.EDU's internet address is 128.197.2.1, so it is host 1 on subnet 2 of Boston University's Class B network 128.197.

Supporting unmodified hosts

If a system does not support subnetting then it will think that other subnets are directly reachable through its own interface; the host will not realize it needs to forward packets destined for other subnets through a gateway. To fix this problem we can use a trick involving ARP *(Address Resolution Protocol.)*

ARP is a protocol designed originally for Ethernets to separate knowledge about the IP addressing and the lower-level Ethernet addressing, which are distinct (that is, an IP address on a local ether network must be mapped to a corresponding Ethernet hardware address.)

The way this is done is that a host broadcasts on the local net requesting the Ethernet address corresponding to an IP address. Normally, the host with that IP address responds with its own Ethernet address.

Motivations for this were to avoid requiring an administrator to maintain tables of IP to Ethernet address mappings and to remain robust even if the hardware address changes (for example, if the Ethernet board is changed due to hardware failure.) Hosts normally maintain a cache of Ethernet to IP address resolutions so this does not have to be done for every packet.

A host *not* supporting subnets will think a remote host on another subnet can be found by broadcasting an ARP request. One trick that is used is to write a program (or add code to the operating system kernel) which responds to such requests with a lie. The lie is that the Ethernet address returned does not correspond to the host but to an appropriate gateway for that destination's subnet.

The host will not verify this response so sends its packets to that gateway and the gateway, observing that the destination IP address is not for itself but for a host on its other side, will forward the packets appropriately.

Proxy ARP

This is known as *Proxy ARP* or "The ARP Hack" in some circles. Notice that it is not necessary that the host providing the Proxy ARP packets be the gateway itself (although this is common and gateway vendors often provide Proxy ARP facilities), it only has to point other hosts towards an appropriate gateway.

The Boston University environment

Boston University is a large, geographically spread out, urban University. Our network must span many buildings and be connected across streets and alleys over which we have little or no control. It would be very difficult for many reasons to use *one* simple network for the entire campus. The campus is currently made up of about a dozen physically distinct networks serving over 200 hosts. They all share one Class B address broken up into subnets.

Our networks are composed of a single Proteon fiber-optic network snaking through the entire campus with Ethernets running off this spine connecting departmental hosts. This is accomplished via Proteon gateways which always have at least one fiber-optic interface and one Ethernet interface. We also have several gateways to the outside world, all located on one central Ethernet in the computing center. To the Arpanet (and other nets) we appear as one network, 128.197.0.0. That single number is all the information that is required in anyone's routing table outside the campus.

Internally we are divided into subnets such as 128.197.2.0 which serves the Math and Computer Science backbone, 128.197.20.0 serving the Information Technology staff, etc. The network mask is 255.255.255.0, the first three octets are the network address (two for the Class B address and one more for the subnets) and the final octet is the host number. We have several systems which do not yet support subnets. These rely on Proxy ARP from both our Proteon gateways and Sun servers.

Choosing subnet parameters

In theory, choosing a correct subnetting scheme seems confusing, in practice it's usually very simple. The main concern is choosing a scheme which provides enough bits for both networks and hosts. Starting with a Class B network number this usually leads one to consider the easy, one octet each, choice of about 256 subnets with 256 hosts on each network or some small modifications, such as 128 networks with 512 hosts each.

continued on next page

Subnet Implementation *(continued)*

Acquiring host and gateway software

Few Ethernets grow to more than a few dozen hosts. The hardware to split a group of hosts onto their own Ethernet (typically based on work groups) is relatively inexpensive and it becomes desirable to try to limit the amount of traffic on each Ethernet by splitting networks like this, based upon expected traffic patterns. It's reasonable to assume that, for example, a group of chemists will mostly exchange network data with their fellow chemists, workstations with their file servers, etc. A more critical decision is the traffic pattern through certain key gateways, particularly those leading towards external networks such as the Arpanet or centralized resources such as a local supercomputer or on-line library database.

Similarly, few organizations need more than a few dozen distinct networks. Boston University, with over 20,000 students and corresponding staff and faculty, is doing fine with about one dozen internal subnetworks, we could double that in the near future but that would still present no problem in terms of subnet addressing. Consequently, 256 networks with 256 hosts each seems more than adequate for the foreseeable future for most organizations, but you will need to do your own projections to confirm this.

One important thing to remember when purchasing systems intended to live in a subnetted environment is to ask the vendor if their software supports subnets! Don't take it for granted. Put it on your list of requirements. It may not be a critical element of your choice of system vendor, but it needs to be known in advance.

Many vendors support subnets, most of the others who don't currently have concrete plans to do so in the near future. If they don't have any support or plans and don't have any good reasons for omitting subnets I would tend to get a little suspicious. Investigate the quality of their overall networking software; do they take networking seriously? Do I need someone who takes networking seriously for this installation? What else don't they support that most vendors do support?

If the vendor can only provide near-term plans for subnet support and you will have to live for some period of time without it, then you may want to investigate the impact this will have on your users. It might very well mean, in the worst case, that users won't be able to get to or from any part of your network not on the local subnet.

If your external gateways (e.g., to the Internet) are also on other pieces of your subnetwork they won't be able to use those gateways either. If the external gateways are on the local subnet you'll be getting these wisecracks about being able to log in to sites thousands of miles away but (possibly) not down the hall.

Mail can usually be configured easily to work around this problem, just set it up to forward all local mail to a host on the same subnet which understands subnets; it might be a subnet gateway but that's not necessary.

To work around all problems you will need to look into Proxy ARP for a host or dedicated gateway on your network. A version for SunOS 3.4 that I wrote is available for anonymous FTP from BU-IT.BU.EDU in file `misc/proxyd.sun3.4.tar`.

You should also find out if any other vendors for hosts or gateways on your subnet support Proxy ARP if you need it.

Hosts which support a 4.3BSD style `/etc/arp` administration command can also be used as a limited form of Proxy ARP on a per-host basis by listing published ARP table entries with (host, gateway-Etheraddress) pairs.

Summary

Subnetting an organization is relatively easy but does require understanding what the mechanism achieves and doing some advanced planning both for the current situation and potential growth. After studying the current network diagram for your site and making reasonable projections for growth over the next several years the first task is to choose an appropriate subnet mask.

The next chore is to go through your host list and ascertain that they all can support subnetting. If not, then other plans will have to be made (or the consequences evaluated.)

Finally, assign subnet values to the existing and projected networks and rearrange host numbers within these subnets to match this new scheme.

A typical way to handle final changes is to declare a "Flag Day," a day on which all hosts, gateways and networks will be changed simultaneously. This is hard to avoid, if only part of the network is changed over then it won't be able to speak to the other parts. For example, at Boston University we announced our intentions that on a given Sunday evening we would change over and nothing would be available for a few hours. We had several experienced staff members come in and, after changing a small piece of the network to assure us, we installed pre-prepared host and network tables throughout the network. Many systems won't easily pick up these new numbers without rebooting (e.g., mail daemons will already have read the old host tables), so a short burst of coordinated effort might be necessary to get everything working again. I don't remember any disasters.

References

Plummer, D., "An Ethernet Address Resolution Protocol," RFC 826.

Mogul, J., "Internet Subnets," RFC 917.

Postel, J., "Multi-LAN Address Resolution," RFC 925.

Clark, D., "A Subnetwork Addressing Scheme," RFC 932.

Karels, M., "Another Internet Subnet Addressing Scheme," RFC 936.

Gateway Algorithms and Data Structures Task Force (GADS), "Toward an Internet Standard Scheme for Subnetting," RFC 940.

Mogul, J. & Postel, J., Internet Standard Subnetting Procedure," RFC 950.

Carl-Mitchell, S. & Quarterman, J., "Using ARP to Implement Transparent Subnet Gateways," RFC 1027.

BARRY SHEIN recently joined Encore Computer Corp. as Manager of Integrated Systems Products. This article was written while he was Manager of Special Projects for the Distributed Systems Group at Boston University. Previously he spent several years at the Harvard School of Public Health working on medical systems applications on the UNIX operating system.

Subnets, Routers, and Bridges

by Fred Ball, Ford Motor Company
Scientific Research Laboratories

Introduction

This article describes the use of Internet Protocol (IP) subnets [1] and routers, and MAC level [2] bridges in a local area/wide area network (LAN/WAN) installed at the Ford Motor Company Research and Engineering (R&E) Center in Dearborn, Michigan. Emphasis is on: 1) the need for multiple subnet masks within the same network, 2) how to use them with commercially available network hardware and software, and 3) how to create invisible sub subnets using bridges with custom filtering capabilities.

Network topology and configuration

The R&E network, a hybrid of baseband and broadband coaxial, fiber optic, and twisted pair cable systems, services 21 R&E buildings on 721 acres. The local area networks in two of the research buildings are Ethernet baseband with a fiber optic link between them. Other buildings in the R&E Center have LANs based on broadband cable systems. Many of these buildings use an Ethernet channel on the campus-wide broadband system for inter-building communications. This network interconnects many heterogeneous devices (e.g., IBM personal computers, various Digital Equipment Corp. systems, Sun and Apollo CAE/CAD engineering workstations, and others) running several different networking protocols. Leased lines, operating at 9,600, 19,200, 56,000, and 1.544 million bits per second and X.25 packet switched circuits, connect other Ford facilities outside the immediate R&E Center area and worldwide. A variety of commercially available gateways, routers, and bridges support this geographically dispersed heterogeneous network.

The Ford R&E network is characterized by its:

1) *Large size*—By the early 1990's the R&E network is expected to have more than 10,000 workstations locally and another 5,000 at external Ford Engineering facilities worldwide.

2) *Geographic dispersion*—Ford has Engineering facilities not only in Southeastern Michigan, but in North and South America, Europe, Australia, and the Far East. The Ford R&E Wide Area Network is expected eventually to include all these areas and will use a combination of leased lines and the Ford owned X.25 packet switched network.

3) *Variety of protocols*—The TCP/IP protocol suite presently is the network of choice for new engineering workstations. There are also large DECnet and Primenet networks. XNS is used mostly for connecting Apollo workstations to Ethernet. There is a small ISO protocol based network in the Scientific Research Laboratories. Other protocols are used in a separate Ford Office Automation network.

4) *Multiple cable types*—Most of the R&E Center building LANs and the Campus LAN are broadband coaxial cable systems. The Scientific Research Laboratories (2 buildings) use baseband Ethernet coaxial cable with a fiber optic cable between them. Coaxial and twisted pair cable systems provide access from the main building systems to the desk.

5) *Variable LAN sizes*—A small engineering CAE/CAD LAN may consist of 2 diskless workstations and one fileserver/router. Some of the larger R&E Center buildings eventually may have up to 2,000 workstations on a single building LAN.

6) *Changing traffic patterns*—Traffic flow is dependent on the workload distribution. Engineering responsibilities for Ford product may be localized within a region (e.g., Europe or U.S.) or shared across regions. Responsibilities may change throughout a product cycle or when new vehicle design efforts are initiated. Major traffic flows are typically local and to/from the Dearborn area.

Although most networks do not exhibit all these characteristics, they *do* share many common problems. Some require solutions that are not common practice such as the use of: 1) multiple subnet masks in IP networks and 2) bridges with custom filtering features that create invisible IP sub subnets.

Effect of subnet size on configuration and cost

The choice of IP subnet size impacts network topology in terms of physical arrangement and both equipment and maintenance costs. For example, consider the three story building wired for a baseband Ethernet network that supports multiple network protocols and 800 workstations in Figure 1. Two segments on each floor use repeaters to connect to the vertical segment that spans all three floors and serves as the building network backbone. A simple bridge provides access to a campus LAN. A large subnet size of 2048 nodes would support multiple buildings and would not require any changes to this configuration.

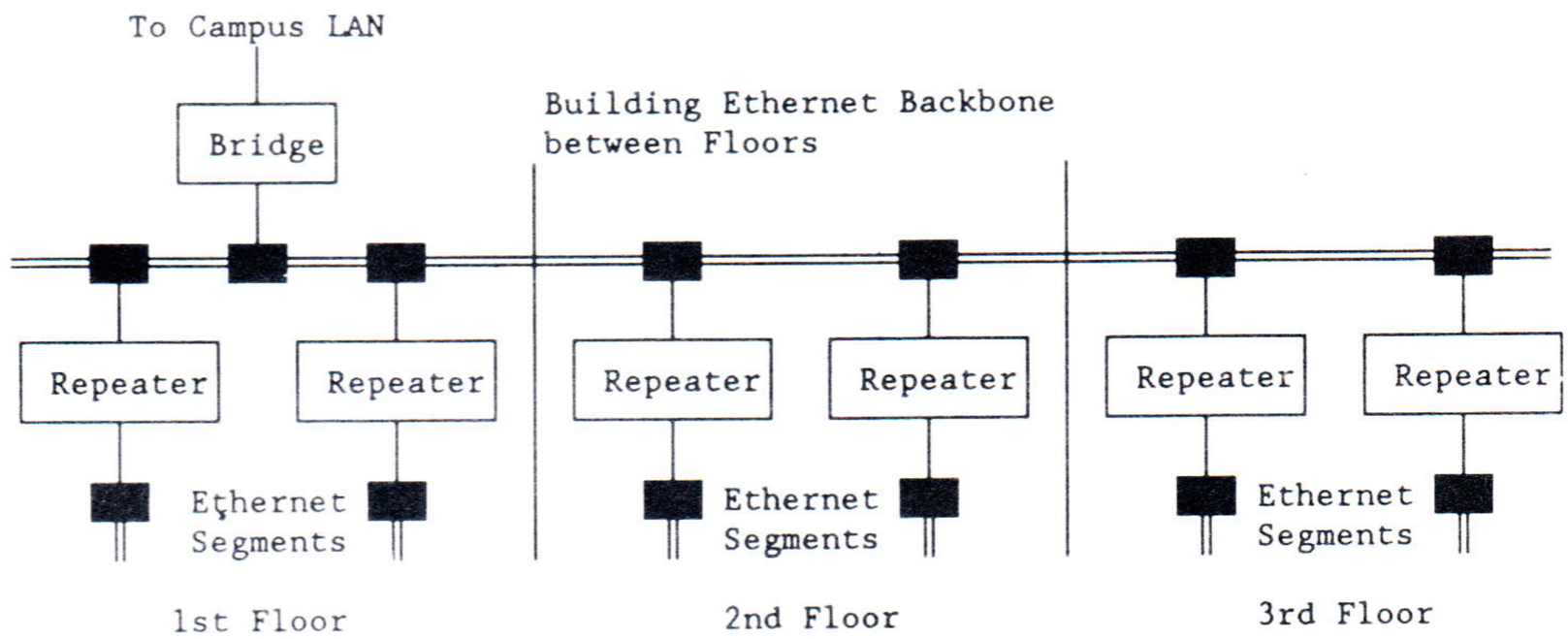


Figure 1: Example building with 800 workstations

A smaller subnet size of 256 nodes requires additional equipment, Figure 2. One possible configuration is to add an IP router and a bridge with custom filtering capabilities on each floor. A second bridge is needed on the third floor because of the fourth IP subnet. The routers support four IP subnets. The bridges are needed for the protocols that are not handled by the routers and must be able to filter the IP packets to prevent routing loops. Not only has this choice of subnet size dictated the network configuration, but it has increased equipment cost by about $85,000 and the resources needed to manage the network.

continued on next page

Subnets, Routers, and Bridges *(continued)*

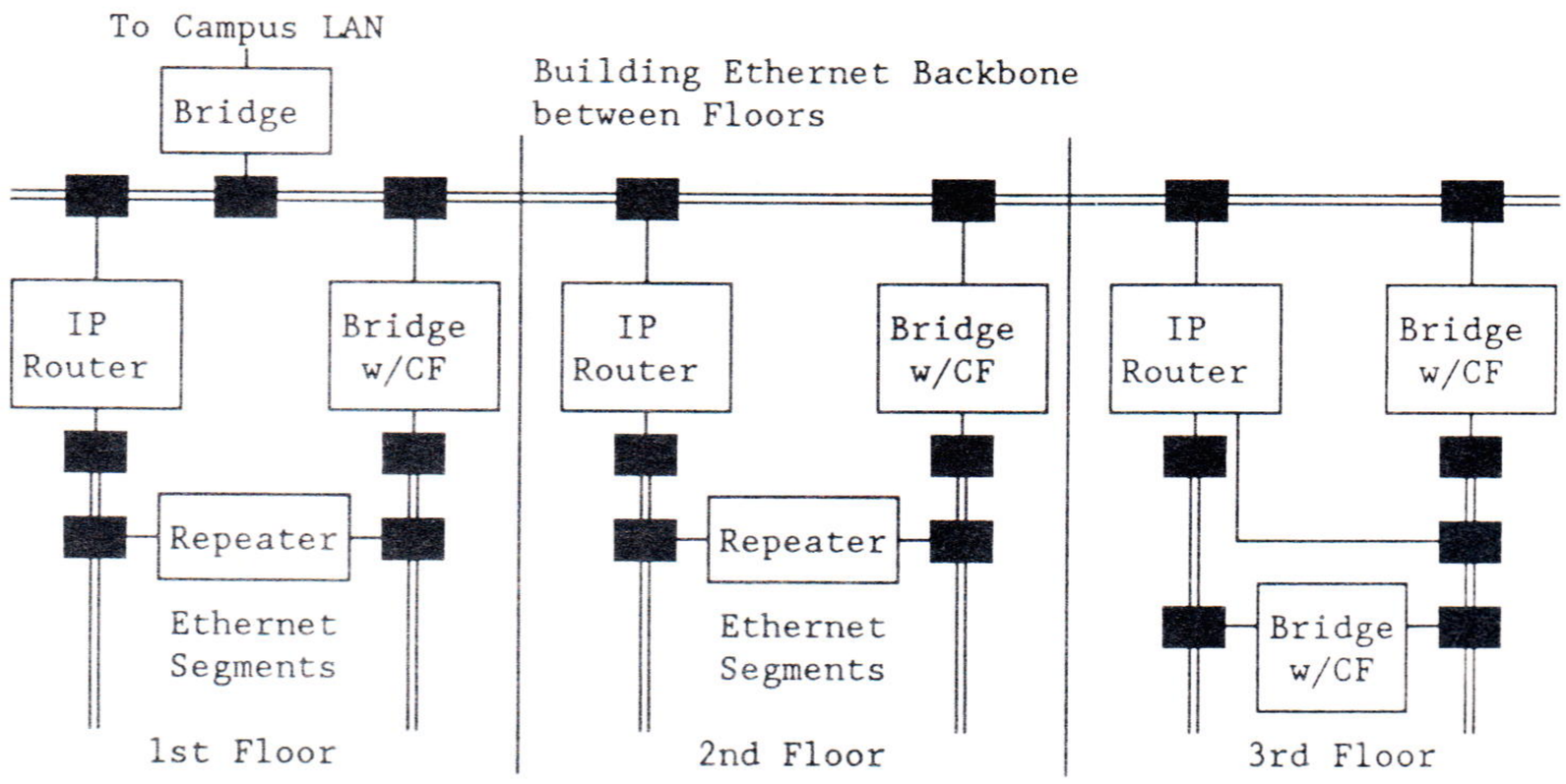


Figure 2: Building upgraded to support a subnet size of 256 nodes

The appropriate choice of subnet size varies according to facility and workstation arrangements. It is highly desirable to be able to use multiple subnet sizes in a large network to provide flexibility in configuration and to conserve addresses.

Multiple subnet masks

The Ford Class B network originally used an 8 bit subnet mask (i.e., 255.255.255.000), resulting in 254 subnets of 254 hosts each. This configuration is not suitable for the R&E network because of the need for larger subnets. However, Ford Aerospace (FAC) facilities outside of Dearborn already have several of these subnets in operation. The present solution is to use multiple subnet masks. The R&E network uses a 3 bit subnet mask (e.g., 255.255.224.000) to divide the network into six usable subnets of 8190 hosts each: 1) 128.5.32, 2) 128.5.64, 3) 128.5.96, 4) 128.5.128, 5) 128.5.160, and 6) 128.5.192. And FAC continues to use its address space of subnets 128.5.32—128.5.63 because they overlap the R&E subnet #1 (128.5.32).

Major computer and workstation vendors were recently queried concerning the advisability of implementing multiple subnet masks on the Ford Class B network, 128.5. All responded that their systems would not support this requirement. Some even recommended that it not be attempted. The problem was how to implement multiple subnet masks when neither the specification, RFC 950 [1] nor the vendors supported it. Due to limited resources, the solution had to be commercially available, reasonably priced, and "easy" to implement.

Careful study of the specifications and available IP router implementations indicated that certain configurations would support multiple subnet masks without requiring special software or hardware. The Ford R&E network presently is divided by using cisco Systems routers installed at the Ford Scientific Research Laboratories (SRL) in Dearborn and at the Ford Aerospace Corporation (FAC) facilities in San Jose, California. The routers are connected by a 9600 bps leased line. Each end uses its own subnet mask as previously described. Dynamic routing has been disabled for this serial link to block the passing of inconsistent routing information between the routers. All the other connections on the two routers are configured with dynamic routing enabled where appropriate. Static routing entries are used to direct the correct routing of packets, according to IP destination address. Specific information on how to configure the routers is available in the *Gateway Server Reference Manual* [3].

Figure 3. shows the present router configurations. The Dearborn router forwards packets with IP addresses: 1) 128.5.32.0 through 128.5.63.255 over the serial line towards San Jose, 2) 128.5.192.0 through 128.5.223.255 over the Ethernet interface, and 3) all others to the Internet router 128.5.192.1. In San Jose, the router sends packets with IP addresses: 1) 128.5.32.0 through 128.5.32.255 over the Ethernet interface, 2) 128.5.33.0 through 128.5.63.255 over the serial interfaces according information provided by RIP or *Routing Information Protocol* [4], and 3) all others over the serial line towards Dearborn.

This configuration works for the Ford network. However, this approach is not usable as a general solution because it does have the following disadvantages: 1) multiple circuits between the R&E and FAC networks are not supported, and 2) the number of static routing entries can be significant.

The present IP standards do not address the problem of variable sized subnets within a users network. A single subnet mask severely restricts the configuration options available to Class A and Class B networks. In fact, under the present rules, large users could get more benefit from using the combination of a Class A or Class B network and several Class C networks. Another approach to this problem is to partition large subnets into smaller divisions without using subnet masks.

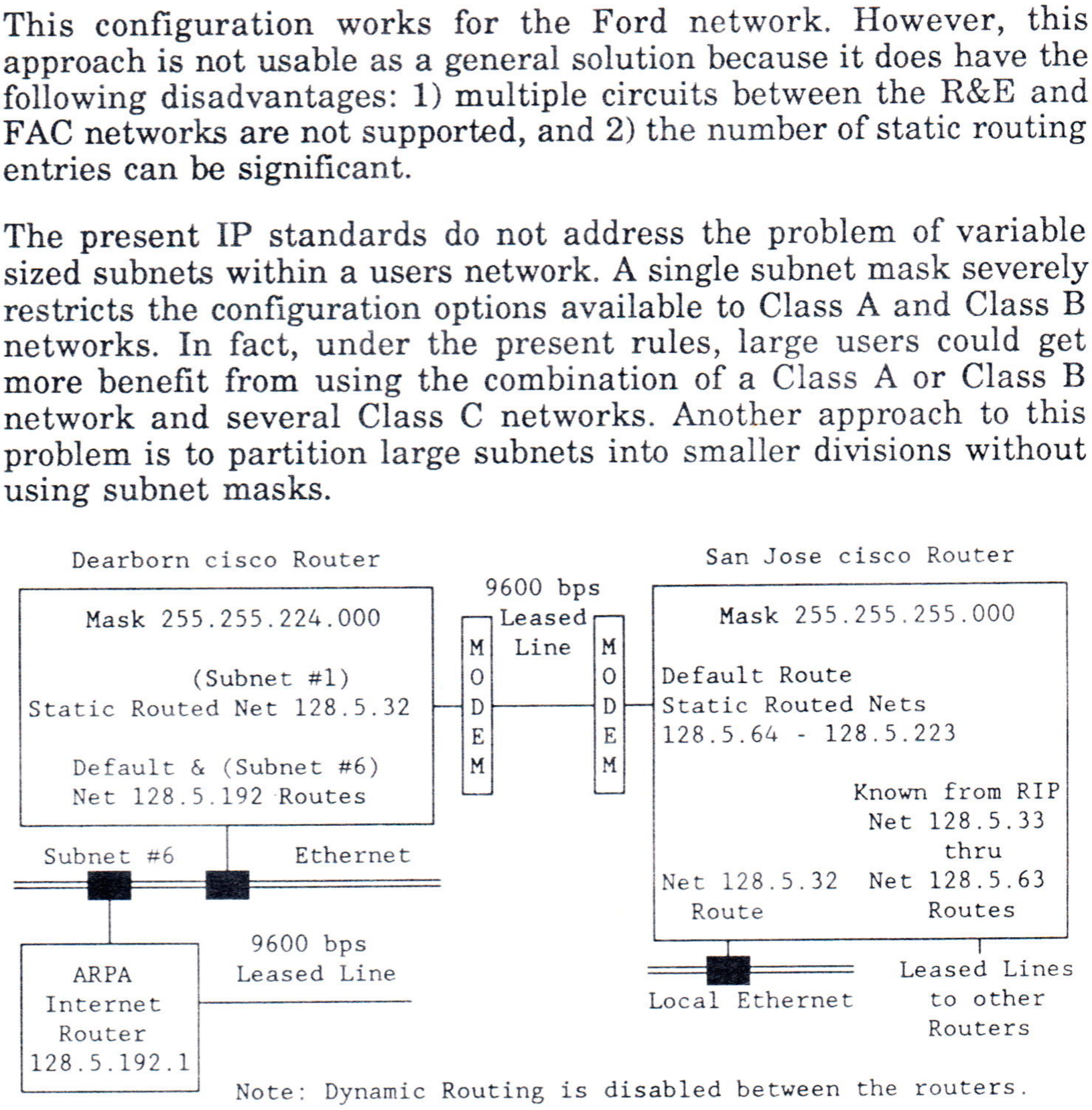


Figure 3: Router configuration

Invisible Sub Subnets

A bridge with custom filtering capabilities can have the characteristics of a simple protocol independent router and may be used to subdivide a large subnet without using an IP router. If a 3 bit subnet mask is used on a Class B network, the individual subnets support 8190 hosts. Many would argue that it is not practical to implement a subnet this large because of the added traffic and processing load from all the broadcast packets. This is generally true unless provisions are made to further subdivide the subnet into *sub subnets*.

For example the Ford subnet #6 (IP addresses 128.5.192.0 through 128.5.223.255) could be divided into four sub subnets of 2048 hosts and allocated to four different buildings within the R&E network, as shown in Figure 4. Each building would be connected to the R&E Campus broadband network using sophisticated bridges that make subnet #6 appear to be on one Ethernet covering the four buildings.

continued on next page

Subnets, Routers, and Bridges *(continued)*

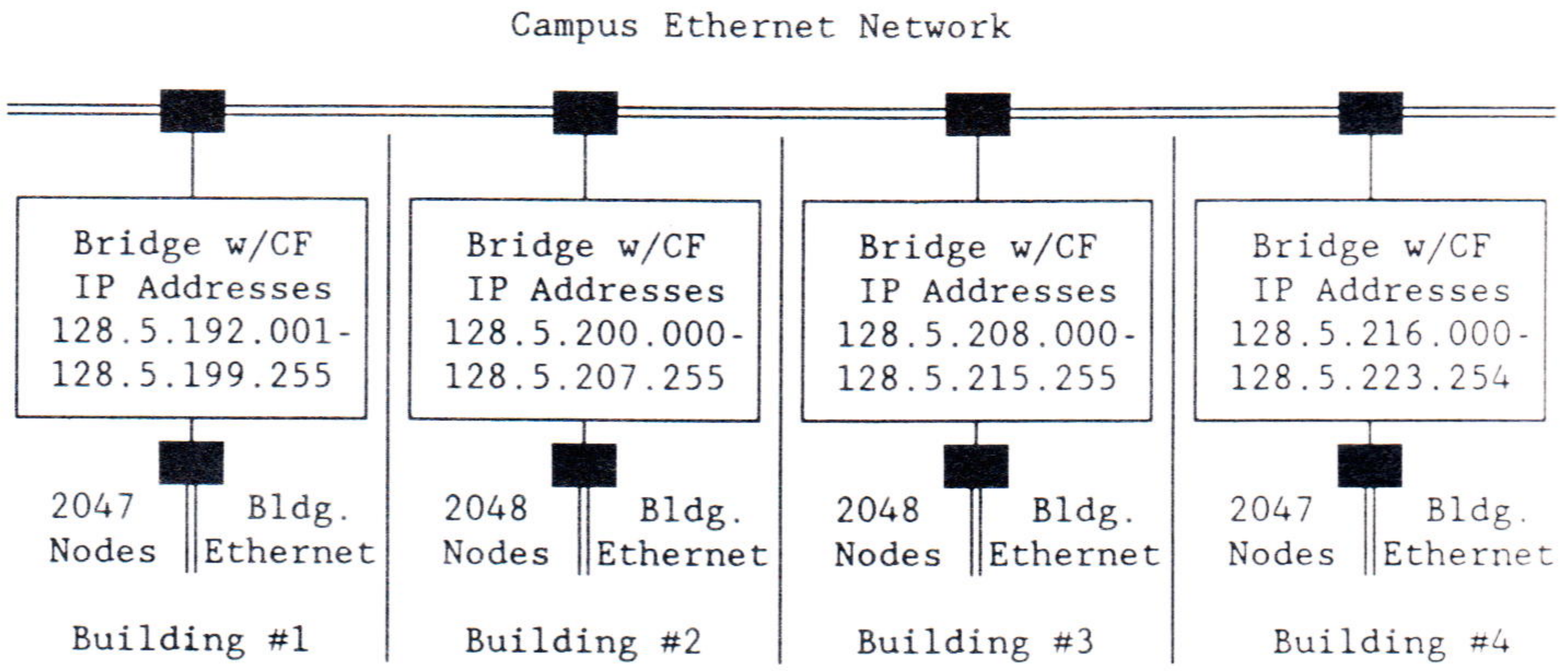


Figure 4: Example 8190 node Subnet bridges with custom filtering

The keys to making this arrangement work are to select a bridge that provides custom filtering and to allocate the IP addresses in blocks to each of the buildings.

The bridge must be able to selectively filter packets based on packet type and ranges of both source and destination IP addresses. The idea is to use the bridge as a simple router to contain the local traffic including local broadcast messages within the respective buildings. For example, the broadcast packets produced by the *Address Resolution Protocol* (ARP) [5] to determine the Ethernet address for a specific IP address are passed by simple bridges, but could be filtered, Figure 5. ARP packets with the source and destination IP addresses on the local sub subnet are not forwarded from the building to Campus LAN. And the ARP packets originating outside the building and destined to one of the other three buildings are not forwarded from the Campus LAN to the local sub subnet. As a result, the sub subnet structure is only known to the bridges and is invisible to all the nodes on subnet #6. The bridge also may be able to handle special routing for protocols other than TCP/IP. The Ford Scientific Research Laboratories presently are experimenting with this approach to determine its operational feasibility.

Field (bytes)	Content Description	Value
1 - 6	Ethernet Destination Address	FFFFFFFFFFFF
7 - 12	Ethernet Source Address	08000200A368
13 - 14	Ethernet Type Code	0806
15 - 20	Various	
21 - 22	ARP Request Opcode	0001
23 - 28	Various	
29 - 32	IP Source Address	8005C007
33 - 38	N/A	
39 - 42	IP Destination Address	8005C009

Figure 5: Example ARP Broadcast packet

Conclusion Multiple IP subnet masks could provide the flexibility to configure complex user networks, but are not a standard. Invisible sub subnets require bridges with custom filtering, but may be another way to deal with the need for flexibility to support multiple protocols and subnet sizes.

References

[1] Mogul, J. & Postel, J., "Internet Standard Subnetting Procedure," RFC 950.

[2] "IEEE Standards Board, Carrier Sense Multiple Access with Collision Detection (CMSA/CD) IEEE Std 802.3-1985," The Institute of Electrical and Electronics Engineers, Inc., 1985.

[3] "cisco Systems, Gateway Server Reference Manual, Revision B," cisco Systems, January, 1988.

[4] Hedrick, C., "Routing Information Protocol," RFC 1058.

[5] Plummer, D., "An Ethernet Address Resolution Protocol," RFC 826.

FREDERIC J. BALL is a Telecommunications Specialist at the Ford Motor Co. Scientific Research Laboratories in Dearborn, Michigan. His responsibilities include the planning and implementation of the Scientific Research Laboratories' network that services over 800 users. He advises other corporate activities on networking issues and problems and coordinates Ford's access to the Internet. Mr. Ball holds an M.B.A. from the University of Michigan, an M.S.E.E. from Wayne State University, and a B.S.E.E. from Michigan State University.

BSD Networking Software now available

The University of California at Berkeley recently announced the availability of the first release of the BSD networking software. It consists of the standard user level applications, (along with their manual pages and some related documentation) and some kernel and **C** library support. It should be noted that this software has only been tested for compilation and operation on 4.3BSD and 4.3BSD-tahoe.

The TCP and IP code is approximately the same as that recently made available via the Arpanet and USENET. Several new algorithms are used in TCP, in particular Van Jacobson's *Slow Start* and dynamic window size selection algorithms and Phil Karn's modification to the roundtrip timing algorithm. These changes increase throughput and reduce congestion and retransmission. Several fixes were made in the handling of IP options and other gateway support.

This software suite is copyright The Regents of the University of California and may be freely redistributed. No previous license, either AT&T or Berkeley is required. The release costs $400.00 US. To request an order form, please contact the distribution office by phone at 415-642-7780, or by email at bsd-dist@ucbarpa.berkeley.edu or uunet!ucbarpa!bsd-dist, or by U.S. Mail at:

CSRG, Computer Science Division
University of California
Berkeley, CA 9472

The US Domain

by Jon Postel,
USC Information Sciences Institute

Introduction

The US domain is an official top-level domain in the *Domain Name System* (DNS) of the Internet community. It is registered with the Network Information Center at SRI International (SRI-NIC). The domain administrators are Jon Postel and Ann Westine at the Information Sciences Institute of the University of Southern California (USC-ISI).

The US domain hierarchy is based on political geography, that is, the US domain is subdivided into states, then cities, and so on. Any computer in the United States may be registered in the US domain. There is no cost for registering a host in the US domain.

Typical host names in the US domain are:

VIXIE.SF.CA.US
DOGWOOD.ATL.GA.US
KILLER.DALLAS.TX.US
GRIAN.CPS.ALTADENA.CA.US

Registration

Because many computers in the United States are already registered in the COM, EDU, and other top level domains, relatively few computers are currently registered in the US domain. The computers that are registered are primarily owned by small companies or individuals (and often located in homes). It is expected than many more computers of all types and belonging to all sizes of organizations will be registered in the US domain.

There is no change in the procedures for registration in, or operation of, other top-level domains such as COM, EDU, GOV, INT, MIL, NET, or ORG. These domains are *not* being moved under the US domain.

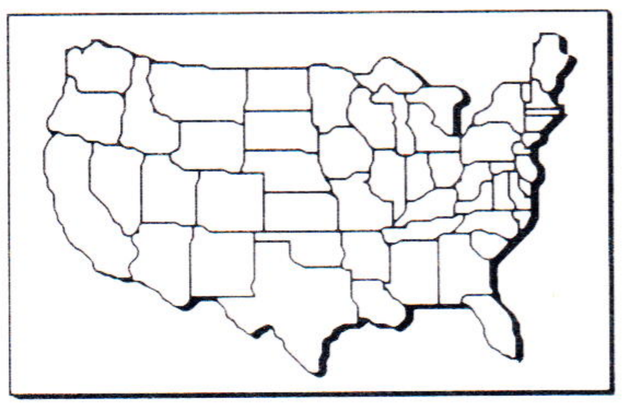
Registration of a host in the US domain does not grant permission to use the Internet or its component networks. Any restrictions on sending mail through (or other use of) the Internet is independent of host registration in the US domain. Registration in the US domain does not allocate any IP address, or cause registration in the HOSTS.TXT file maintained by SRI-NIC.

Currently, the US domain and all of its subdivisions (that is, states and cities) are managed by the US Domain Administrator. At some time in the future the administration of individual states and cities will be transferred to appropriate responsible people.

The administrator of a company or the organizer of a group (or "domain park") of users with individual hosts may coordinate the registration of the group by forwarding all the information for the group to the US Domain Administrator.

Use of MX records

The explicit specific information for each host must be provided. All fully qualified names must be unique. If a host is not directly on the Internet an MX record is required pointing to an Internet host for forwarding. The forwarding host must be directly on the Internet (that is, have an IP addresss), no "double MX-ing" is allowed.



A group coordinator of, for example, the Computer Club in Chicago (CCC), could arrange to coordinate the registration of all the computers used by members of the club. The registered names might have the form:

PC37.CCC.CHI.IL.US MX 10 CS.UOFC.EDU

Only hosts on the Internet can act as forwarding hosts. Hosts on systems such as CSNET, UUCP, BITNET, must be registered with an Internet forwarding host. When registering a destination host in the US domain with an MX record, the requester is responsible for also registering the destination host with the administrator of the forwarding host. For example, when a messages is sent to "Susan@PC37.CCC.CHI.IL.US" it will be routed to the Internet host "CS.UOFC.EDU" as directed by the MX record. The host "CS.UOFC.EDU" must know some way of delivering the message to the host "PC37.CCC.CHI.IL.US" (*uucp*, SLIP, whatever). So the destination host (PC37.CCC.CHI.IL.US) must be known to (registered with) the forwarding host (CS.UOFC.EDU), as well as being registered in the DNS database.

The administrator of the destination host must make an agreement with the administrator of the forwarding host for the forwarding service. This agreement must be in place before the request for registration is sent to the US Domain Administrator.

A section of the DNS database is called a "zone." With careful coordination, a domain (like EDU) can be divided into several zones. This has been done for the EDU and COM domains to aid in the registration of hosts from the UUCP, CSNET and BITNET communities. If a host is registered in UUCP, BITNET, or CSNET zone (as something.EDU or something.COM), it need not be registered in the US domain, unless a geographical name (something.city.state.US) is desired.

Only *one* name

It is the policy that a computer must have a single primary name, so it should *not* be registered in both US and COM (or both US and EDU). It is possible to have "nicknames" for a brief period while a host name change is in progress. Wildcard records are not currently allowed in the US domain.

The US domain is currently supported by four name servers:

VENERA.ISI.EDU, VAXA.ISI.EDU, HERCULES.CSL.SRI.COM, and NNSC.NSF.NET.

Further information

For information on Internet domains in general, see RFC 1034, Mockapetris, P., "Domain Names—Concepts and Facilities," and RFC 1035, Mockapetris, P., "Domain Names—Implementation and Specification." For more information about the US domain please contact Ann Westine at WESTINE@ISI.EDU.

JONATHAN B. POSTEL is Director of the Systems Division of the Information Sciences Institute of the University of Southern California. Jon has been involved in the development of computer communication protocols and applications from the early days of the Arpanet. His current interests include multimachine internetwork applications, multimedia conferencing and electronic mail, very large networks, and very high speed communications. Jon received a BS and MS in Engineering and a PhD in Computer Science from the University of California, Los Angeles.

Upper Layer Functionality: A Letter to the Editor

by Helmut Hildebrandt, Stuttgart Institute of Technology

I read with some interest an article by Mr. Vitaliano of VXM Technologies, Inc., entitled *Upper Layer Interoperability*, in a recent issue of *ConneXions*. [Ed.: Volume 2, No. 10, October 1988]. Although I agree with the need for a session layer capability across various protocol suites, I strongly disagree with the solution proposed by Mr. Vitaliano.

In short, it is suggested that "TIM," a product of VXM Technologies, Inc., be used to supply TCP/IP-based networks with a session layer capability. The reasons given for this is that the TIM session provides "a continuous, bidirectional, asynchronous (non-blocking) data stream," and that RPC mechanisms, such as Sun Microsystems' RPC, do not. Unfortunately, Mr. Vitaliano confuses RPC mechanisms with session layer capabilities. To understand this issue, I draw your attention to 2 articles by Marshall Rose of Wollongong, which readers of *ConneXions* are no doubt familiar with.

RFC 1006

The first article is entitled *ISODE: Horizontal Integration in Networking*. [Ed.: Volume 1, No. 1, May 1987]. In this article, the mechanism defined in RFC 1006, *ISO Transport Services on top of the TCP*, is introduced. This allows any OSI application to use a TCP/IP-based network for transport purposes—in such a way that the application believes that it is operating in a "pure" OSI network. As such, the application uses the OSI session layer to achieve session layer functionality over TCP/IP.

ROS

In the second article, entitled *Building Distributed Applications in an OSI Framework*, a semi-automated approach to using the OSI Remote Operations Service (ROS) was described. [Ed.: Volume 2, No. 3, March 1988]. ROS ultimately uses the OSI session services, and, if the RFC 1006 mechanism is being employed, then runs over a TCP/IP-based network. One should note that the ROS is inherently "a continuous, bidirectional, asynchronous (non-blocking) data stream," thereby fulfilling Mr. Vitaliano's definition of session layer capabilities. Of course, ROS is the result of many years of work in the International Standards community, and is used in various OSI applications such as Message Handling, Directory Services, and remote database access. This is in contrast to TIM, which is a vendor-proprietary protocol. [Ed.: See note at end of article].

I must stress that the ROS approach easily achieves the claimed advantages of the TIM approach:

> "... organizations can get rapid turn-around time for development of network or distributed applications. They avoid the wasted time and effort of trying to figure out the mysteries of TCP/IP interfacing."

Transition

But, there is one important advantage over the TIM approach: because a "real" OSI application is developed under this scheme, and because a "real" OSI session protocol is used, any application built, tested, and matured in a TCP/IP-based network can be moved to an OSI-based network with absolutely *no* coding changes. All one need do is re-link the application with an interface to a "real" OSI transport layer, and the application will run—unaltered.

Openly available

If, as Mr. Vitaliano suggests, OSI is "the final outcome for computer networks," then it seems clear that the RFC 1006 mechanism is the optimal method for implementing an OSI session capability in TCP/IP-based networks. Best of all, an implementation of all of the software developed for this approach is openly available, as mentioned in the first article I cite.

[Editor's note: *ConneXions* has contacted Mr. Vitaliano regarding the proprietary nature of the TIM system. Mr. Vitaliano comments that his company, VXM Technologies, has plans to make TIM an open protocol by specifying it in an RFC in the near future].

HELMUT M. HILDEBRANDT is a Senior Researcher at the Stuttgart Institute of Technology in the Federal Republic of Germany, where he works on computer languages in support of OSI. He received his D.Sc. degree in Electrical Engineering from the Technical University München, in 1961.

Book Reviews

Internetworking with TCP/IP—Principles, Protocols and Architecture by Douglas Comer, Prentice-Hall, ISBN 0-13-470154-2. This book is designed to be a comprehensive introduction to the TCP/IP protocol suite. Comer successfully manages to explain almost every aspect of TCP/IP networking, from how packets are routed to how hostnames get looked up. The book is intended both as an introduction for the advanced undergraduate and as a reference for professionals. Often that constitutes and unhappy mix of readers: the undergraduates get buried by technical details while the professional finds little intellectual substance amidst the introductory text. Comer, however, manages to make this mix work. The text is easy to read and avoids the mathematics and heavy technical jargon that frustrates the beginner; at the same time, it offers the professional a useful reference that at least touches on all aspects of TCP/IP networking. The bibliography is good, and at the end of each chapter the reader is pointed towards additional reading. —*Craig Partridge*

Computer Networks, Second Edition by Andrew S. Tanenbaum, Prentice-Hall, ISBN 0-13-162959-X. This book presents an excellent reference for computer communications. Although the outline of the book roughly follows the OSI model, Tanenbaum uses the model as a framework rather than a bible. For example, he discusses remote procedure calls as a session layer facility (where RPC is found in traditional systems), rather than as a part of the application layer (where RPC is found in OSI). Nor is the book devoted entirely to OSI, some aspects of TCP/IP are discussed along with considerable attention to issues independent of protocol suites, such as network security. The second edition has been extensively updated to accurately reflect networking technology as it was in 1988: it is almost entirely a different book from the first edition. The timely nature of the material plus the tremendous breadth and insight of Andrew Tanenbaum make *Computer Networks* THE computer-communications reference book for the 90's. —*Marshall T. Rose*

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

PUBLISHER Daniel C. Lynch

EDITOR Ole J. Jacobsen